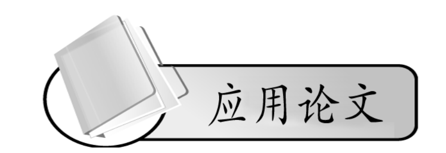


ng-info-chart：基于自定义HTML标签的交互式可视化组件*

陈　挺[1]　王小梅[1]　吕伟民[1, 2]
[1](中国科学院文献情报中心　北京　100190)
[2](中国科学院大学　北京　100049)

摘要：【**目的**】设计并实现基于模型–视图–控制器(MVC)前端AngularJS框架的可视化组件ng-info-chart。【**应用背景**】优秀的情报分析平台往往需要使用多个复杂的可视化图谱组合展示分析结果，需要更有效地构建复杂的、支持大量交互操作的网页端情报分析可视化图谱。【**方法**】ng-info-chart集成多种可视化图谱，使用AngularJS自定义扩展标签统一封装，通过自定义HTML标签直接在页面中调用绘图方法。【**结果**】ng-info-chart可视化组件随着研究团队情报分析项目不断深入与完善，现已集成5个第三方可视化类库中11种可视化图谱，支持IE9+、Firefox等主流桌面浏览器。【**结论**】利用可视化组件实现数据异步获取、自动检测数据变化与实时图谱绘制等功能，极大简化了情报分析系统中复杂可视化图谱的开发工作。

关键词： 可视化　前端MVC　自定义扩展标签
分类号： G250

1　引　言

随着大数据时代的到来，支持情报分析的可视化不再是简单的图表统计所能满足的，而是采用多源数据并使用复杂可视化图谱发现数据中隐藏的关联关系和发展趋势。为满足情报分析需求，情报产品开发时，复杂的、用户频繁交互的可视化图谱被大量使用，如陈超美博士开发的CiteSpace计量分析软件[1]和专注网络计量分析的Pajek分析软件等[2]，均为用户提供揭示复杂关联关系的可视化分析图谱，并支持用户交互、筛选形成最终深度分析结果。这些可视化分析软件均是客户端程序，而网页端的复杂情报可视化工具并不丰富，以简单地统计分析图表为主。究其原因主要是复杂可交互可视化分析对服务器、浏览器压力过大，编码复杂，开发难度大。随着信息技术发展，特别是新的浏览器引擎技术的高速发展，浏览器端脚本JavaScript的运算速度有了极大提高，可以较短时间内绘制复杂的可视化图谱。本文调研了业界最新的网页端可视化图谱开发模式与开发框架，并与情报研究分析需求结合，设计并实现基于自定义HTML标签调用的可视化组件ng-info-chart，方便开发人员调用可视化图谱，并将可视化数据与用户交互操作双向绑定，简化了大量Ajax异步数据获取与调用多种可视化类库的操作，实现智能监测数据变化并自动更新图谱功能。

2　情报分析中可视化需求与相关技术

2.1　情报分析中的可视化需求

信息可视化不仅用视图来显示多维的非空间数

通讯作者：陈挺，ORCID：0000-0003-4785-5367，E-mail：chent@mail.las.ac.cn。
*本文系国家自然科学基金项目“科学结构特征及其演化动力学分析方法应用研究”(项目编号：71173211)的研究成果之一。

据，使得用户加深对数据含义的理解，而且可以用形象直观的图像指导服务过程，从而改善服务的效率和效果[3]。当今情报分析工作中所需的可视化图谱越来越复杂，情报分析人员对分析图谱的可操作性的要求也越来越高，例如 Global Open Access Development 平台①中，开放获取数据地图可视化图谱除了展示相关数据统计外，还需在地图中直接筛选、计算、检索、显示/隐藏相关数据。如在重大科技动态图谱项目②中，多维对比图中需通过时间轴拖曳与重要度得分动态筛选数据，并根据不同的统计需求对可视化图的统计模式和横纵坐标轴进行转换，再如科研竞争力全向评估系统③中，采用了力矢量布局图展示论文、项目的布局，用户可以直接在页面中调整聚类参数、力矢量布局系数、筛选分析对象等，系统根据用户交互操作形成最终可视化分析结果。虽然现今基于 JavaScript 的可视化组件有很多选择，但集成多种类库绘制图谱开发存在难度大、维护成本高等问题，开发一个支持用户大量交互，功能完备的可视化分析平台需要投入大量人力物力，为此急需一种松耦合、模块化的可视化图谱的开发模式。本研究对业界常用可视化类库和前端开放框架进行详细的调研，设计并初步实现可视化组件库 ng-info-chart。

2.2　常用网页端可视化类库

开源程序仓库 GitHub 提供的编程语言趋势分析中可以看出，近年来 JavaScript 在网页端的使用量快速增长[4]，同时基于 JavaScript 的前端可视化开发类库也得到了快速发展，逐渐成为网页可视化的首选[5]，有大量优秀的可视化工具和图谱可供选择。而其他诸如 Flash、Silverlight、Java Applet 等网页端可视化技术渐渐远离开发人员的视线。

网页端可视化图谱的开发中，针对不同可视化分析需求有多种 JavaScript 可视化工具和类包可供选择，但面对复杂的情报分析平台，单一的可视化分析工具往往无法满足全部的可视化分析需求，常常需要调用多个可视化分析工具和类库。如果可视化需求极为特殊，无法直接调用已有封装的图谱，更需要底层定制图谱。表 1 为情报分析平台开发时常用可视化分析工具类库对比，其中列举的 6 个工具和类库，可以满足从简单统计图表到力矢量图、时序分析图和地理信息图等深度可视化分析需求，表 1 第三列为程序员问答网站关于可视化工具提问数统计，从提问量中可以看出，Highchat.js、D3.js 的社团活跃度远远超过 Echarts.js 和 Datav.js，因此本文可视化工具集成也优先选用 Highchat.js 和 D3.js。

表 1　常用可视化图谱比较(统计于 2015 年 11 月)

可视化 js 类库	面向可视化需求	Stack OverFlow 提问次数
Highchats.js	绘制折线图、饼图、柱状图、树型图、散点气泡图、雷达图等常规分析图表。浏览器兼容性好，API 功能完善，社区活跃。	25 874
Echarts.js	百度公司的开源可视化工具，相比 Highcharts，图表类型更丰富，不但有折线图、饼图、柱状图等常见可视化图表，还有热力图、河流图、字符云、力矢量图等新型可视化图谱。	4
D3.js	基于 SVG 矢量图技术的绘图框架，与其他可视化类库不同，D3.js 不是基于已有的可视化图谱的封装，而是一种面向数据的可视化高级语言。主要用于实现复杂的可视化图谱，如力矢量图、河流图等。	32 698
Datav.js	Datav.js 是淘宝下可视化实验室开源的可视化库，利用 D3.js 将常用的可视化图谱进行封装，方便开发者调用。	0
Google map	在开发基于地理信息位置的分析图谱时，如果面向全世界，免费的类库没有太多的选择，只有 Google map 和 Open map，Google map 的社区非常活跃，通常是开发者的首选。	80 270

2.3　前端开发框架 AngularJS

最常用的 jQuery Ajax 异步数据更新再调用第三方可视化图谱的开发模式代码过于繁杂，无法满足复杂的支持交互的情报分析平台开发需求，本研究

①http://www.grc-oa.org.cn/.

②http://strategy.las.ac.cn/strategy/index.php/nav/rbreak.

③http://strategy.las.ac.cn/strategy/index.php/nav/rcompete.

需要一种高效前端可视化图谱开发模式，调研业界主流前端 MVC 框架后发现，虽然近年来前端 MVC 框架层出不穷，但是经过对比，Google 公司发布的 AngularJS[6]是其中最有影响力的前端 MVC 框架。选择一个开源技术框架，除了技术本身的优劣外很重要的一个评判标准是开源社区的活跃度，从图 1 可以看出业界三大前端 MVC 框架中 AngularJS 无论从开发人员的关注度还是技术社区活跃度都远远超过其他框架，因此本文选用 AngularJS 作为前端可视化组件的开发框架。

Metric	AngularJS	Backbone.js	Ember.js
Stars on Github	40.2k	18.8k	14.1k
Third-Party Modules	1488 ngmodules	256 backplugs	1155 emberaddons
StackOverflow Questions	104k	18.2k	15.7k
YouTube Results	~93k	~10.6k	~9.1k
GitHub Contributors	96	265	501
Chrome Extension Users	275k	15.6k	66k
Open Issues	922	13	413
Closed Issues	5,520	2,062	3,350

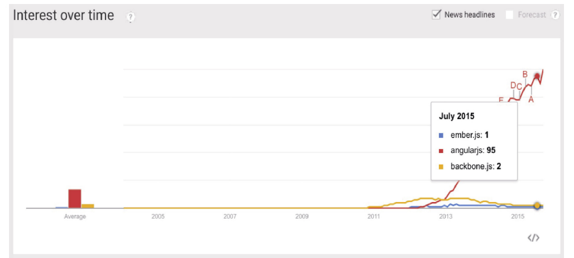


图 1　三大主流前端 MVC 框架 GitHub 活跃度和 Google 检索关注度统计[7]

AngularJS 是 Google 公司开发并维护的纯客户端 JavaScript 技术的前端框架，用于扩展、增强 HTML 功能、简化开发过程以及测试复杂度，专为构建强大的 Web 应用而设计[8]。框架为开发人员提供了大量先进功能: MVC 结构、数据双向绑定、路由、模块化管理、自定义扩展标签(Directive)、依赖注入、前端 HTML 模板等，其中 AngularJS 独有的“自定义扩展标签”、“双向数据绑定”功能可以很好地满足复杂可交互情报分析图谱的技术需求，大大简化用户交互而产生的 Ajax 数据操作和可视化方法的调用。同时框架也提供“模块化管理”以及“依赖注入”等高级语言的特性弥补原生 JavaScript 脚本语言在模块化调用中的不足。

(1) 自定义扩展标签

自定义扩展标签是 AngularJS 框架提供的自定义 DOM 元素标记，允许开发人员扩展 HTML 标签并将动态数据与之绑定[9]。在可视化组件设计中，可通过自定义扩展标签将不同的可视化类库封装成不同的 HTML 标签，并直接在 HTML 页面中通过自定义标签调用可视化方法。

(2) 双向数据绑定

双向数据绑定(Two Way Data Binding)或许是 AngularJS中最重要且无法通过jQuery简单实现的功能之一，借助它可以将一个私有作用域中的属性同DOM 中的属性值进行绑定[10]。一个典型的Web应用程序中大部分的前端代码都是DOM元素监听，获取、更新数组与对象。借助双向数据绑定功能可以简化前端数据的操作，无论是控制器(Controller)中已保存的数据，还是从接口API中获取更新数据，或用户操作修改了数据，绑定数据会自动在整个应用各个层中自动同步，包括自定义标签中调用的数据，非常适合面向大量交互的可视化图谱的开发。

3　ng-info-chart 可视化组件设计与实现

3.1　复杂可视化图谱开发中的问题

使用原生 JavaScript 或 jQuery 实现交互式可视化图谱开发往往包括多个环节: Ajax 数据获取，调用可视化类库与 div 元素绑定，渲染可视化数据，交互事件(如用户点击、拖曳按钮)与 Ajax 数据获取再次绑定，更新可视化数据，移除已有可视化图谱，重新渲染新可视化图谱。复杂的可视化图谱，常在同一页面中调用多组可视化数据并调用多个可视化类库，渲染多个可视化图谱，或者可视化数据与多个用户操作绑定，容易造成前端 JavaScript 程序开发难度大，多种外部依赖库调用混乱，后期代码难以维护等问题。

3.2　可视化组件设计

考虑到以上问题，ng-info-chart 可视化组件设计时，利用 AngularJS 框架中独有的“自定义扩展标签”与“双向数据绑定”特性，在可视化组件中设计并实现了以下 6 个功能以简化可视化图谱开发:

(1) 可视化类库模块化封装: 可视化组件采用模块化封装，利用 AngularJS 框架中的模块管理功能①，将第三方可视化方法分别封装入不同组件模块

(AngularJS Module)中，调用组件绘制图谱时无需在页面中载入全部的可视化组件，只需调用对应模块即可，提高运行速度。

(2) 外部资源动态引入：如果分析系统中需要使用多个可视化类库，则需要开发者在页面中载入多个类库对应文件，载入资源的版本以及载入的顺序均非常容易造成图谱渲染错误，可视化组件将库及相关资源载入工作从页面中移入可视化组件统一载入与管理，开发人员无需在页面中管理多种资源的载入，只需载入可视化组件文件 infochart.js 即可。

(3) 定义图谱的"自定义扩展标签"：不再使用监测 div 元素 id 的方式绑定可视化图谱与页面元素，而是将不同的可视化类库封装成不同的"自定义扩展标签"，如使用 D3.js 开发的力矢量图封装成"ng-info-force"标签并定义其对应传入的可视化数据，在 HTML 页面中通过自定义标签直接调用，这样简化图谱与页面元素和数据绑定工作，提高了程序的可读性和可维护性。

(4) "双向数据绑定"可视化数据：将可视化数据存储在 AngularJS 模型(Model)层中，不论用户修改数据还是 API 中更新数据，模型层都会自动更新最新数据，可视化组件自动继承模型中存储的最新数据，开发人员无需将更新数据与可视化方法反复对应。

(5) 常用可视化图类绘图方法封装：将多个情报分析平台常用 js 类库绘图方法与常用工具如点击，悬浮在标签中封装，方便调用。

(6) 自动监测数据变化并刷新图谱：在可视化组件中监控数据变化，一旦发现用户操作改变了数据或者数据接口更新数据，组件将自动刷新可视化图谱。

可视化组件调用流程图如图2所示，可视化图谱绘图方法通过自定义扩展标签统一封装，每个标签中还包括调用外部可视化类库资源方法、监测数据变化方法。自定义扩展标签根据第三方可视化类库不同，封装在不同的Angular模块中，按照本身可视化图谱名称统一命名，如封装Highchart.js中的热力图方法(Heatmap)的自定义扩展标签命名为ng-chart- heatmap。调用可视化绘图时先通过依赖注入Highchart绘图模块，再在HTML页面中引入标签<ng-chart-heatmap>绘制热力图谱，组件调用详见3.4节。

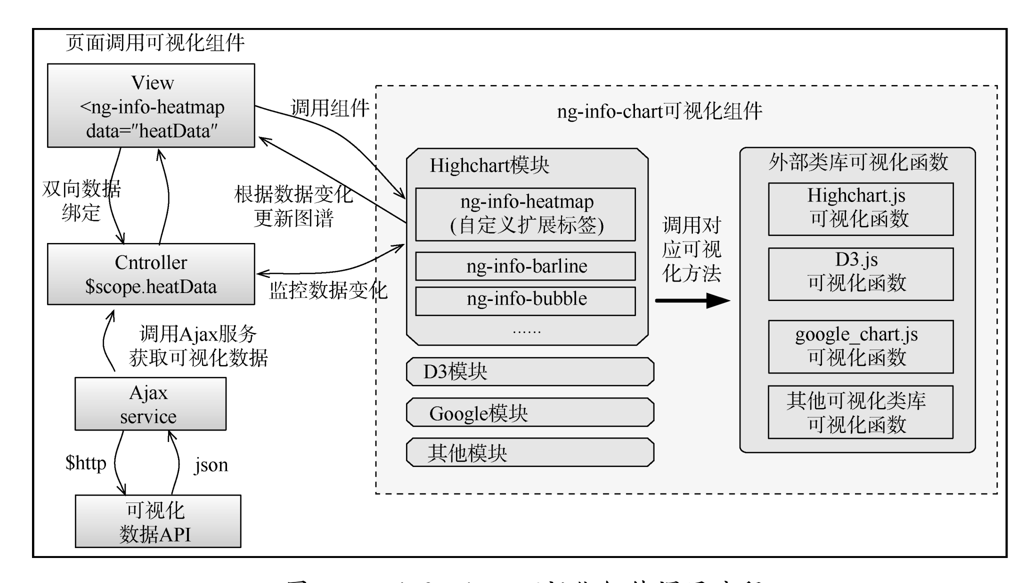


图 2　ng-info-chart 可视化组件调用流程

3.3　可视化类库在自定义扩展标签中的封装

为了将多个第三方可视化类库有序地封装入 AngularJS 框架的自定义扩展标签内，组件开发时遵从 AngularJS 框架的最佳实践开发模式[9]。可视化组件结构如图 3 所示，组件底层为多个第三方可视化类库，将其中常被情报分析平台使用的可视化图类方

①https://docs.angularjs.org/guide/module.

法封装在 AngularJS 框架的服务[①](Service Factory)中，需要时可通过依赖注入的方式引入自定义扩展标签中调用，在自定义标签中统一设计自定义调用标签名称与定义对应绘图数据，并通过在自定义标签中调用"$watch"命令监控数据更新并自动刷新可视化图谱。

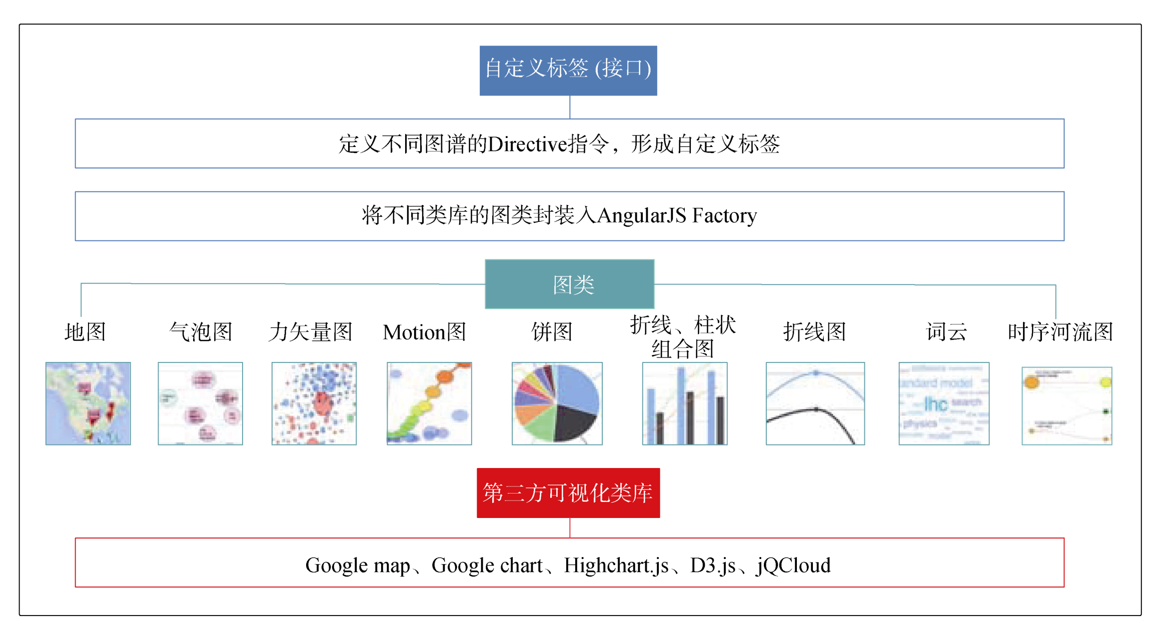


图 3　ng-info-chart 可视化组件结构

以热力图为例，自定义标签封装 JavaScript 代码如下：

(1) 可视化图谱方法模块化封装：将第三方可视化图谱绘图方法封装入 AngularJS 框架中的 Factory 中，以供自定义标签中调用，一个模块可以封装多个 Factory，Factory 中可含有多个绘图方法，同一个类库的方法可以封装在一个 Factory 中，核心 Factory 代码如下所示：

```
/*使用 Factory 封装 Highchart 中部分可视化图谱*/
var chart = angular.module("chart", []);
/*Heatmap 图谱*/
chart.factory('highchart', function(){
    return {
        heatmap: function(container, data){
            /*此处 Highchart 绘图方法*/
        },
      /*封装更多 Highchart 绘图方法*/
        barline: function(container, data){
        ....
        },
    }
});
```

(2) 封装自定义扩展标签<info-chart-heatmap>：将标签封装 infoChart.highcharts 模块中，并引入之前封装的绘图模块 Chart 中的 Highchart Heatmap 的绘图方法，同样一个模块可包含多个标签，核心代码如下所示：

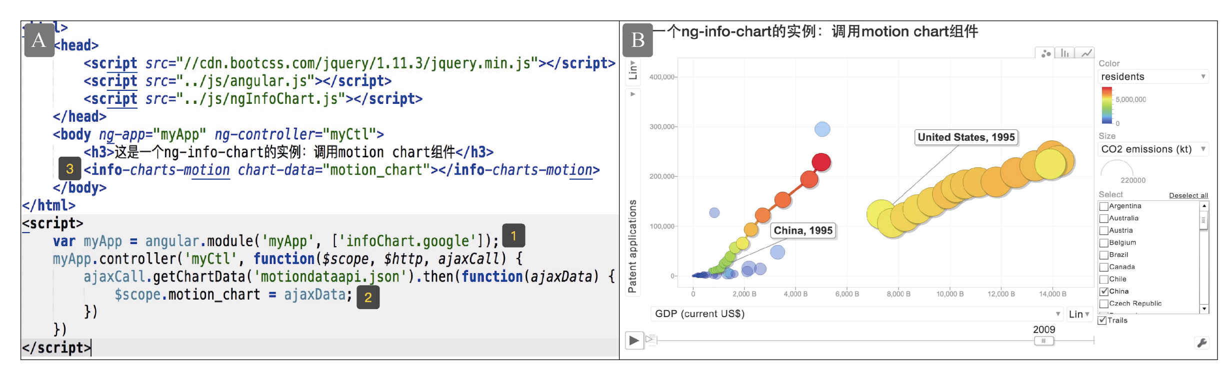
```
/*定义新模块 infoChart.highcharts, 封装自定义扩展标签*/
var infoChartHighcharts = angular.module("infoChart.highcharts",
  ['highcharts']);
/*定义"自定义 html 标签"-infoChartsHeatmap, 在其中引入对应
  的绘图方法: highchart*/
infoChartHighcharts.directive('infoChartsHeatmap',
  function($window, highchart){
    return{
        restrict:'EA',
        template:"<div class='myCharts'></div>", // 定义标签
          中 html 容器
        scope:{
            chartData:'='//获取标签属性中 chartData 中的数据
        },
        link: function(scope, elem, attrs){
            // 使用$watch 方法监控数据变化, 如果数据有变
              化自动更新图谱
            scope.$watch('chartData', function(nv){
```

①https://docs.angularjs.org/guide/services.

```
                    var container=elem.find('div');
                    highchart.heatmap(container, nv);
                })
            }
        }
    });
```

3.4　ng-info-chart 可视化组件调用

相比直接调用第三方可视化类库，通过ng-info-chart组件绘制图谱在代码层面更易实现。图4以调用可视化组件绘制多维动态对比图(Motion Chart)为例说明可视化组件调用流程，通过AngularJS依赖注入调用可视化组件中Google可视化模块，如图4(a)中标签1的位置；通过Ajax服务从后端数据接口获取可视化数据，并将数据储存入Model $scope.motion_data中，如图4(a)中标签2的位置；此时开发人员只需在HTML页面中使用自定义标签"<ng-info-motion data=motion_Data></ng-info-motion>"调用可视化组件，如图4(a)中标签3的位置；并将组件所需可视化数据通过标签属性"data=motion_data"将Model中存储数据与组件关联，组件会自动在HTML页面中的标签位置绘制多维动态图谱如图4(b)所示。之后无论是数据接口更新还是用户交互操作修改可视化数据，可视化组件会自动监测数据变化并重新动态生成可视化图谱。相比于传统的开发模式调用第三方可视化类库，使用ng-info-chart开发代码更简洁易读，开发更为高效，尤其便于后期维护。



(a) 使用ng-info-chart组件绘制多维对比图示例　　　　(b) 可视化图谱效果展示

图 4　可视化组件调用流程示例

其中，图4(a)的标签1为通过依赖注入引入绘图所需方法模块；标签2为在JavaScript中调用Ajax异步服务，获取绘图数据并储存入Model $scope.motion_chart中；标签3为在HTML页面中，通过自定义HTML标签<ng-info-motion>调用可视化组件并载入Model中的可视化数据。通过以上三个步骤，可视化组件会调用底层封装的第三方可视化类库绘多维对比图。

4　应用实例

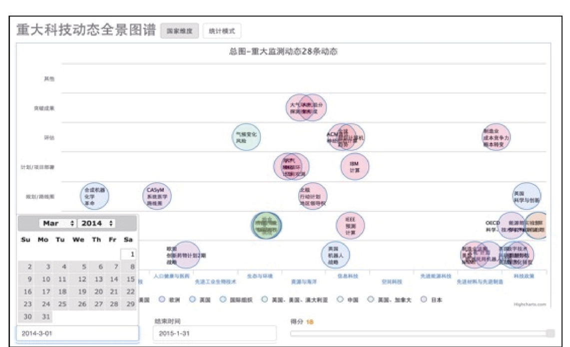
ng-info-chart可视化组件起始于以往情报分析平台对复杂可视化图谱的实际需求，从2014年初至今已集成5个第三方可视化类库中的11种可视化图谱(见表2)。有统计饼图、柱状图等基础分析图表，也有如气泡图与Motion图的多维对比分析图，还有基于Google map的地理信息位置分析图谱和解释关系的力矢量图和河流时序图。并在多个情报研究项目中实际应用(见图5)，如开放获取平台中的交互地图，重大科技动态全景图谱项目的多维对比气泡图，科研竞争力全向评估项目中的力矢量图，科学结构地图网页版中的河流时序图等。相比直接在原生JavaScript或者jQuery中调用第三方可视化类库，使用ng-info-chart组

表 2　ng-info-chart 已集成第三方可视化类库列表

第三方可视化类库	图类	封装的自定义扩展标签
Highchats.js	饼图	ng-info-pie
	折线图	ng-info-line
	柱状图	ng-info-bar
	折线柱状图	ng-info-linebar
	气泡图	ng-info-bubble
	热力图	ng-info-heatmap
D3.js	力矢量图	ng-info-force
	河流时序图	ng-info-sankey
Google map	地图	ng-info-map
Google chart	Motion-chart	ng-info-motion
jQCloud.js	词云	ng-info-cloud



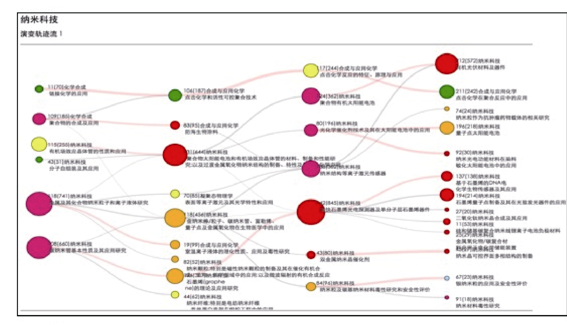
(a) 开放获取平台中的交互地图



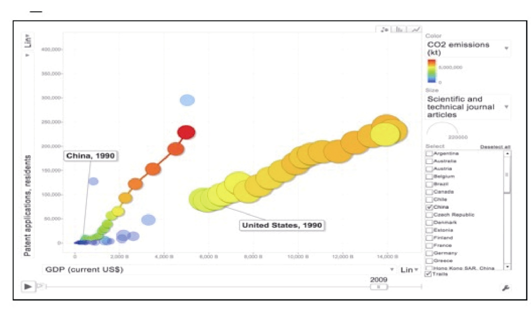
(b) 重大科技动态全景图谱项目的多维对比气泡图



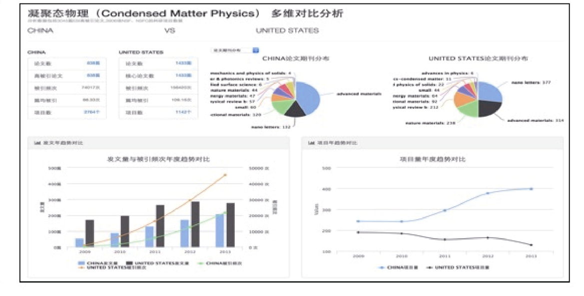
(c) 科研竞争力全向评估项目中的力矢量图



(d) 科学结构地图网页版中使用的河流时序图



(e) 战略信息集成服务平台中内嵌的图 (多维动态图)



(f) 常用的饼图、折线图、折线柱状图

图 5　ng-info-chart 可视化组件在情报分析平台中实际应用场景

件绘制图谱大大减少了前端代码，开发效率更高，同时代码更简洁易读，便于后期维护。同时 ng-info-chart 可视化组件还在不断完善中，根据不同的可视化需求，逐渐补充封装更多可视化组件。现有可视化组件均支持 IE9+、Chrome、Firefox 等主流桌面浏览器。

5　结　语

ng-info-chart 可视化组件是一个非常实用的可视化集成工具，它满足了在网页端松耦合、模块化绘制复杂的、多种类型和支持频繁交互的可视化需求，大大简化了新型情报分析平台的中复杂的可视化图谱的开发工作。过去的两年中支持了多项情报分析平台、工具的开发。ng-info-chart 可视化组件现有图谱种类有限，目前还不能支持全部的可视化需求，但是随着情报研究工作的不断深入，本研究团队将不断扩展组件的内容，希望在未来可以服务更多情报研究需求。

参考文献：

[1] Chen C. CiteSpace: Visualizing Patterns and Trends in Scientific Literature [CP/OL]. [2015-08-10] http://cluster.cis.drexel.edu/~cchen/citespace/.

[2] Batagelj V, Mrvar A. Pajek——Analysis and Visualization of Large Networks[A] // Graph Drawing [M]. Springer-Verlag Berlin Heidelberg, 2004:77-103.

[3] Kirk A. Data Visualization: A Successful Design Process [M].

Packt Publishing, 2012.

[4] La A. Language Trends on GitHub [EB/OL]. [2015-08-30]. https://github.com/blog/2047-language-trends-on-github.

[5] 赵越, 陈志伟, 蔡淑惠, 等. 大数据量科学计算数据的动态 Web 可视化[J]. 现代计算机, 2012(5): 3-6. (Zhao Yue, Chen Zhiwei, Cai Shuhui, et al. Web Visualizing for Large Mount of Scientific Computing Data [J]. Modern Computer, 2012(5): 3-6.)

[6] Darwin P B, Kozlowski P. AngularJS Web Application Development [M]. Packt Publishing, 2013.

[7] Uri Shaked. AngularJS vs. Backbone.js vs. Ember.js [EB/OL]. [2015-08-01]. https://www.airpair.com/js/javascript-framework-comparison.

[8] Jain N, Mangal P, Mehta D. AngularJS: A Modern MVC Framework in JavaScript [J]. Journal of Global Research in Computer Science, 2015, 5(12): 17-23.

[9] Green B, Seshadri S. AngularJS [M]. O'Reilly Media Inc., 2013.

[10] Lerner A. Ng-book: The Complete Book on AngularJS[M]. Fullstack.io, 2013.

作者贡献声明:

陈挺: 设计组件技术方案, 编写程序代码, 起草论文;
王小梅: 设计组件技术方案, 修改论文;
吕伟民: 调研可视化方案, 编写程序代码。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据由作者自存储, E-mail:chent@mail.las.ac.cn。

[1] 陈挺. nginfochart-master.zip. 可视化组件项目源代码仓库链接. https://github.com/jy00295005/nginfochart.

[2] 陈挺. motion_chart.csv. 图 4 绘制动态对比图谱的可视化数据. https://github.com/jy00295005/nginfochart/blob/master/motion_chart.csv.

收稿日期: 2016-02-18
收修改稿日期: 2016-03-22

ng-info-chart: The Visualization Component Based on Customized HTML Tags

Chen Ting[1] Wang Xiaomei[1] Lv Weimin[1,2]
[1](National Science Library, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: [**Objective**] This research designs and implements the ng-info-chart, a front end visualisation component based on the MVC framework AngularJS. [**Context**] A good information analysis system requires multiple complex visualzation charts to present the results. Therefore, we need to create an advanced interactive Web-based visualzation charts for the new systems. [**Methods**] We intergrated visualzation charts with the ng-info-chart and the AngularJS Directive packages. The new component could call the charts Directive directly using a customized HTML tag. [**Results**] The ng-info-chart visualisation component has intergrated 5 third-party visualisation libraries of 11 types of visual charts. It supports IE9+, Firefox and other popular Web browsers. [**Conclusions**] The new visualisation component implements data asynchronization, automatic detection of data change, and real-time online visualzation. It also simpilfies the complex visualzation tasks for the information analysis system.

Keywords: Visualization Front end MVC Dirervtive